A Purely Model-Theoretic Semantics for Disjunctive Logic Programs with Negation *

Pedro Cabalar[1], David Pearce[2], Panos Rondogiannis[3], and William W. Wadge[4]

[1] Department of Computer Science
Corunna University, Spain
cabalar@udc.es
[2] Department of Informatics, Statistics and Telematics
Universidad Rey Juan Carlos, Madrid, Spain
davidandrew.pearce@urjc.es
[3] Department of Informatics & Telecommunications
University of Athens, Athens, Greece
prondo@di.uoa.gr
[4] Department of Computer Science
University of Victoria, Victoria, Canada
wwadge@cs.uvic.ca

Abstract. We present a purely model-theoretic semantics for disjunctive logic programs with negation, building on the *infinite-valued* approach recently introduced for normal logic programs [9]. In particular, we show that every disjunctive logic program with negation has a *non-empty* set of minimal infinite-valued models. Moreover, we show that the infinite-valued semantics can be equivalently defined using Kripke models, allowing us to prove some properties of the new semantics more concisely. In particular, for programs without negation, the new approach collapses to the usual minimal model semantics, and when restricted to normal logic programs, it collapses to the well-founded semantics. Lastly, we show that every (propositional) program has a finite set of minimal infinite-valued models which can be identified by restricting attention to a finite subset of the truth values of the underlying logic.

1 Introduction

The semantics of disjunctive logic programs with negation has been the subject of a number of recent research works [8, 2, 3, 11, 12, 4]. A comparative study of all these approaches easily leads to the conclusion that at present there is no consensus on what is the "right approach" to the semantics of disjunctive logic programs with negation. In other words, the quest for an intuitive and broadly acceptable semantic approach appears at present to be unfulfilled. Motivated by this state of affairs, we introduce in this paper a novel, purely model-theoretic semantics for disjunctive programs, which generalizes both the well-founded semantics of normal logic programs as well as the minimal model semantics of (negationless) disjunctive programs. An important characteristic of the new approach is that it is purely logical: the meaning of a program is characterized solely by examining its models (actually, its *infinite-valued models*, see below).

* Partially supported by the the CICyT project TIC-2003-9001-C02, and by the Greek General Secretariat for Research and Technology and the Spanish MEC under a joint project of Scientific and Technological collaboration between Greece and Spain

Having a purely logical semantics allows one to reason about programs using properties of the logic under consideration.

The present work builds on the *infinite-valued* approach that was recently introduced for normal logic programs [9]. In [9] a new infinite-valued logic is introduced and it is demonstrated that every normal logic program has a unique minimum model under this new logic; moreover, it is shown that when this model is collapsed to three-valued logic, it coincides with the well-founded semantics. It is therefore natural to ask: "can the infinite-valued approach be lifted to the class of disjunctive logic programs with negation giving in this way a respectable new semantics for this class of programs?".

In this paper we reply affirmatively to this question. In particular, we argue that the semantics of a disjunctive logic program with negation can be captured by the program's *set of minimal infinite-valued models* which, as we demonstrate, is always non-empty. Moreover, we show that the infinite-valued semantics can be equivalently defined using Kripke models. This alternative characterization allows us to prove properties of the new semantics more concisely. In particular, we prove that when restricted to programs without negation, the new approach collapses to the usual minimal model semantics, and when restricted to normal logic programs, it collapses to the well-founded semantics. Finally, we demonstrate that every program has a *finite* set of minimal infinite-valued models; actually, these models can be identified by restricting attention to a finite subset of the truth values of the underlying logic. We conclude with a comparison of the proposed approach with some other proposals for assigning semantics to disjunctive logic programs with negation that are related to well-founded semantics.

2 The Infinite-Valued Semantics

In this section we define the infinite-valued semantics for disjunctive logic programs with negation. Our presentation extends the one given in [9] for normal logic programs. We study the class of disjunctive logic programs:

Definition 1. *A* disjunctive logic program *is a finite set of clauses of the form:*

$$p_1 \vee \cdots \vee p_n \leftarrow q_1, \ldots, q_k, \sim r_1, \ldots, \sim r_m \tag{1}$$

where $n \geq 1$ *and* $k, m \geq 0$.

Note that in this paper we consider only finite programs; the results can be lifted to the more general first-order case (with a corresponding notational overhead).

The basic idea behind the infinite-valued approach is that in order to have a purely model theoretic semantics for negation-as-failure, one should consider a richer logical framework than classical logic. Informally, we extend the domain of truth values and use these extra values to distinguish between ordinary negation and negation-as-failure. Consider for example the following (normal) program:

$$p \leftarrow \qquad r \leftarrow \sim p \qquad s \leftarrow \sim q$$

Under the negation-as-failure approach both p and s receive the value *True*. We would argue, however, that in some sense p is "truer" than s. Namely, p is true because there is a rule which says so, whereas s is true only because we are never obliged to make q true. In a sense, s is true only by default. Our truth domain adds a "default" truth value T_1 just below the "real" truth T_0,

and a weaker false value F_1 just above ("not as false as") the real false F_0. We can then understand negation-as-failure as combining ordinary negation with a weakening. Thus $\sim F_0 = T_1$ and $\sim T_0 = F_1$. Since negations can effectively be iterated, our domain requires a whole sequence $\ldots, T_3, T_2, T_1$ of weaker and weaker truth values below T_0 but above a neutral value 0; and a mirror image sequence $F_1, F_2, F_3, \ldots$ above F_0 and below 0. In fact, in [9] a T_α and a F_α are introduced for all countable ordinals α; since in this paper we deal with finite propositional programs, we will not need this generality here.

In [9] it is demonstrated that, over this extended domain, every normal logic program with negation has a unique minimum model; and that in this model, if we collapse all the T_α and F_α to *True* and *False* respectively, we get the three-valued well-founded model [10]. Considering the above example program, its minimum infinite-valued model is $\{(p, T_0), (q, F_0), (r, F_1), (s, T_1)\}$, and therefore its well-founded model is $\{(p, T), (q, F), (r, F), (s, T)\}$. In this paper we extend the results of [9] by demonstrating that *every disjunctive logic program has a (non-empty) set of minimal infinite-valued models.*

The above discussion can now be formalized as follows. We first need to define an infinite-valued logic whose truth domain consists of the following values:

$$F_0 < F_1 < F_2 \cdots < 0 < \cdots < T_2 < T_1 < T_0$$

Intuitively, F_0 and T_0 are the classical *False* and *True* values and 0 is the *undefined* value. The values below 0 are ordered like the natural numbers. The values above 0 have exactly the reverse order. In the following we denote by V the set consisting of the above truth values. A notion that will prove useful in the sequel is that of the *order* of a given truth value:

Definition 2. *The* order *of a truth value is defined as follows: order*$(T_n) = n$, *order*$(F_n) = n$ *and order*$(0) = +\infty$.

Let Q be a set of propositional symbols out of which our programs are constructed. Interpretations are then defined as follows:

Definition 3. *An infinite-valued interpretation is a function from the set Q of propositional symbols to the set V of truth values.*

In the rest of the paper, the term "interpretation" will mean an infinite-valued one (unless otherwise stated). As a special case of interpretation, we will use $\emptyset$ to denote the interpretation that assigns the F_0 value to all members of Q.

Definition 4. *The meaning of a formula with respect to an interpretation I can be defined as follows:*

$$I(\sim A) = \begin{cases} T_{n+1} & \text{if } I(A) = F_n \\ F_{n+1} & \text{if } I(A) = T_n \\ 0 & \text{if } I(A) = 0 \end{cases} \qquad \begin{aligned} I(A \wedge B) &= min\{I(A), I(B)\} \\ I(A \vee B) &= max\{I(A), I(B)\} \\ I(A \leftarrow B) &= \begin{cases} T_0 & \text{if } I(A) \geq I(B) \\ I(A) & \text{if } I(A) < I(B) \end{cases} \end{aligned}$$

The notion of satisfiability of a clause can now be defined:

Definition 5. *Let Π be a program and I an interpretation. Then, I* satisfies *a clause $p_1 \vee \cdots \vee p_k \leftarrow l_1, \ldots, l_n$ of Π if $I(p_1 \vee \cdots \vee p_k) \geq I(l_1 \wedge \cdots \wedge l_n)$. Moreover, I is a* model *of Π if I satisfies all clauses of Π.*

We denote by L_∞ the *infinite-valued logic* induced by infinite-valued models. Given an interpretation of a program, we adopt specific notations for the set of atoms of the program that are assigned a specific truth value and for the subset of the interpretation that corresponds to a particular order:

Definition 6. *Let Π be a program, I an interpretation and $v \in V$. Then $I \parallel v = \{p \in Q \mid I(p) = v\}$. Moreover, if $n < \omega$, then $I\sharp n = \{(p, v) \in I \mid order(v) = n\}$.*

The following relations on interpretations will be used later in the paper:

Definition 7. *Let I and J be interpretations and $n < \omega$. We write $I =_n J$, if for all $k \leq n$, $I \parallel T_k = J \parallel T_k$ and $I \parallel F_k = J \parallel F_k$.*

Definition 8. *Let I and J be interpretations and $n < \omega$. We write $I \sqsubset_n J$, if for all $k < n$, $I =_k J$ and either $I \parallel T_n \subset J \parallel T_n$ and $I \parallel F_n \supseteq J \parallel F_n$, or $I \parallel T_n \subseteq J \parallel T_n$ and $I \parallel F_n \supset J \parallel F_n$. We write $I \sqsubseteq_n J$ if $I =_n J$ or $I \sqsubset_n J$.*

Definition 9. *Let I and J be interpretations. We write $I \sqsubset_\infty J$, if there exists $n < \omega$ (that depends on I and J) such that $I \sqsubset_n J$. We write $I \sqsubseteq_\infty J$ if either $I = J$ or $I \sqsubset_\infty J$.*

In comparing two interpretations I and J we consider first *only* those variables assigned "standard" truth values (T_0 or F_0) by at least one of the two interpretations. If I assigns T_0 to a particular variable and J does not, or J assigns F_0 to a particular variable and I does not, then we can rule out $I \sqsubseteq_\infty J$. Conversely, if J assigns T_0 to a particular variable and I does not, or I assigns F_0 to a particular variable and J does not, then we can rule out $J \sqsubseteq_\infty I$. If both these conditions apply, we can immediately conclude that I and J are incomparable. If exactly one of these conditions holds, we can conclude that $I \sqsubseteq_\infty J$ or $J \sqsubseteq_\infty I$, as appropriate. However, if neither apply, then I and J are equal in terms of standard truth values; they both assign T_0 to each of one group of variables and F_0 to each of another. In this case we must now examine the variables assigned F_1 or T_1. If this examination proves inconclusive, we move on to T_2 and F_2, and so on. Thus $\sqsubseteq_\infty$ gives the standard truth values the highest priority, T_1 and F_1 the next priority, T_2 and F_2 the next, and so on.

It is easy to see that the relation $\sqsubseteq_\infty$ on the set of interpretations is a partial order (i.e., it is reflexive, transitive and antisymmetric). On the other hand, for every $n < \omega$, the relation $\sqsubseteq_n$ is a preorder (i.e., reflexive and transitive).

From the above discussion it should be now clear that *the infinite-valued semantics of a disjunctive logic program with negation is captured by the set of $\sqsubseteq_\infty$-minimal infinite-valued models of the program. In Section 4 we'll see that this set is always non-empty. These ideas are illustrated by the following example:*

Example 1. Consider the program:

$$b \vee l \leftarrow \sim p \qquad l \vee p \leftarrow$$

We examine the minimal models of this program. Obviously, in every minimal model either l or p must have the value T_0 (this is due to the second clause). Assume first that l is T_0; then this immediately gives the minimal model $\{(l, T_0), (p, F_0), (b, F_0)\}$. Assume on the other hand that p is T_0; this implies that $\sim p$ is F_1, and therefore either b or l must be F_1 (or greater). Therefore, we also have the minimal models $\{(l, F_0), (p, T_0), (b, F_1)\}$ and $\{(l, F_1), (p, T_0), (b, F_0)\}$.

We denote by L_∞^{min} the non-monotonic logic induced by $\sqsubseteq_\infty$-minimal L_∞ models.

3 Kripke semantics

We present an alternative but equivalent representation of the infinite-valued semantics in terms of Kripke models. This representation is useful in several respects. First, as a heuristic device it may help in visualising and proving properties about the semantics (Section 5). Second, it may help to relate the semantics to other approaches based on possible-worlds frames, such as equilibrium and partial equilibrium logic ([6, 4]). Third, it may provide a basis in the future when searching for axiomatic systems capturing the underlying logic L_∞.

Definition 10 (Centered linear frame). *A* centered linear frame *is a Kripke frame* $\langle W, \leq \rangle$ *with a set of worlds* W *consisting of two distinguished elements* w_∞, w'_∞ *plus two* ω*-sequences* $w_0, w_1, \ldots$ *and* $w'_0, w'_1, \ldots$ *and a linear ordering* '$\leq$' *satisfying,* $w_i \leq w_{i+1}$, $w_i \leq w_\infty$, $w'_{i+1} \leq w'_i$, $w'_\infty \leq w'_i$ *and* $w_\infty \leq w'_\infty$ *for any* $i < \omega$.

From Definition 10 it follows that $w_i \leq w'_j$ for any i, j. Furthermore, we can depict both infinite chains $w_0, w_1, \ldots$ and $\ldots, w'_1, w'_0$ respectively bounded by w_∞ and w'_∞ in the middle as follows:

$$w_0 \leq w_1 \leq \cdots \leq w_\infty \leq w'_\infty \leq \cdots \leq w'_1 \leq w'_0.$$

Given any world $w \notin \{w'_\infty, w'_0\}$ we define $next(w)$ as the immediate successor of w in the chain, that is, $next(w_i) = w_{i+1}$, $next(w_\infty) = w'_\infty$ and $next(w'_{i+1}) = w'_i$.

Definition 11 (Centered linear model). *A* centered linear model *is a Kripke model* $\langle W, \leq, \sigma \rangle$ *where* $\langle W, \leq \rangle$ *is a centered linear frame and* $\sigma : Q \times W \longrightarrow \{0, 1\}$ *is a valuation such that* $\sigma(p, w) = 1, w \leq u \Rightarrow \sigma(p, u) = 1$ *and* $\sigma(p, w_0) = 0$ *and* $\sigma(p, w'_0) = 1$ *for all atoms* p.

Given a Kripke model, we let W_i, W'_i stand for the sets of atoms that are true at the respective worlds w_i, w'_i, for $i = 0, 1, \ldots, \infty$. From Definition 11 we conclude:

$$\emptyset = W_0 \subseteq W_1 \subseteq \cdots \subseteq W_\infty \subseteq W'_\infty \subseteq \cdots \subseteq W'_1 \subseteq W'_0 = Q \tag{2}$$

where in particular

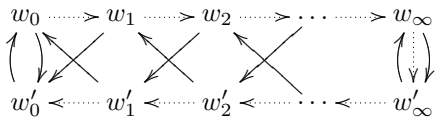
$$\bigcup_i W_i \subseteq W_\infty \text{ and } W'_\infty \subseteq \bigcap_i W'_i.$$

An interesting way of describing a centered linear model M is using the sequence $M = (\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_\infty)$ where each $\mathbf{W}_i$ is a three-valued interpretation $\mathbf{W}_i = (W_i, W'_i)$ so that atoms in W_i, $W'_i \setminus W_i$ and $Q \setminus W'_i$ are respectively seen as *true*, *undefined* and *false* up to order i. We may define the standard "less or equal truth" relation $\preceq$ between pairs so that $\mathbf{W}_i \preceq \mathbf{W}_j$ iff $W_i \subseteq W_j$ and $W'_i \subseteq W'_j$. This allows rephrasing (2) as a simple chain $\mathbf{W}_0 \preceq \mathbf{W}_1 \preceq \cdots \preceq \mathbf{W}_\infty$ with $\mathbf{W}_0 = \langle \emptyset, Q \rangle$ assigning false to all atoms. Interpretation $\mathbf{W}_\infty$ contains the maximum truth in the chain and is important for comparisons with well-founded semantics (Proposition 4 in Section 5). A three-valued interpretation like $\mathbf{W} = (W, W)$ is said to be *complete* (no undefined atoms). We say that $M = (\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_\infty)$ is *i-complete* for some $i \in \{1, 2, \ldots, \infty\}$ when $\mathbf{W}_i$ is complete. Note that this means that $\forall j \ (j \geq i \Rightarrow \mathbf{W}_j = \mathbf{W}_i)$ and $\mathbf{W}_\infty = \mathbf{W}_i$.

Definition 12 (Routley frame). *A* Routley frame *is a triple* $\langle W, \leq, * \rangle$ *where* $\langle W, \leq \rangle$ *is a Kripke frame and* $* : W \to W$ *is such that* $x \leq y$ *if* $y^* \leq x^*$.

Definition 13 (Zigzag model). *A* zigzag model *is a tuple* $\langle W, \leq, *, \sigma \rangle$ *where* $\langle W, \leq, \sigma \rangle$ *is a centered linear Kripke model and* $\langle W, \leq, * \rangle$ *is a Routley frame with* $*$ *defined as:* $(w_i)^* = w'_i$ *and* $(w'_i)^* = w_i$ *for* $i = 0$ *and* $i = \infty$*;* $(w'_{j+1})^* = w_j$ *and* $(w_{j+1})^* = w'_j$ *for all* $j < \omega$.

The structure below shows the effect of $*$ in solid lines, and the linear accessibility relation $\leq$ in dotted lines:



The name of "zigzag" comes from the path followed by successive applications of the $*$-function. Given a zigzag model $M = \langle W, \leq, *, \sigma \rangle$, valuation σ is extended to an arbitrary formula φ by means of the usual conditions for positive connectives in intuitionistic logic, and for negation by the following condition: $\sigma(\sim\varphi, w) = 1$ iff $\sigma(\varphi, w^*) = 0$.

Proposition 1. *For any zigzag model* $\langle W, \leq, *, \sigma \rangle$ *and any formula* φ, $\sigma(\varphi, w) = 1$ *and* $w \leq u \Rightarrow \sigma(\varphi, u) = 1$.

We say that M is a *model* of a theory Γ, written $M \models \Gamma$, if $\sigma(\varphi, w_1) = 1$ for all $\varphi \in \Gamma$ (note that satisfaction is in world w_1 and not w_0).

Definition 14 (Induced valuation). *Given a zigzag model* $M = \langle W, \leq, *, \sigma \rangle$ *we define its* induced valuation *function* $\hat{M}$ *that assigns a value* $\hat{M}(\varphi) \in V$ *to any formula* φ *as follows:*

$$\hat{M}(\varphi) \stackrel{\text{def}}{=} \begin{cases} T_i \text{ iff } w_i \not\models \varphi \text{ and } w_{i+1} \models \varphi \\ F_i \text{ iff } w'_{i+1} \not\models \varphi \text{ and } w'_i \models \varphi \\ 0 \quad \text{iff } w_\infty \not\models \varphi \text{ and } w'_\infty \models \varphi \end{cases}$$

This definition applied to atoms implies $\hat{M} \parallel T_i = W_{i+1} \setminus W_i$, $\hat{M} \parallel F_i = W'_i \setminus W'_{i+1}$ and $\hat{M} \parallel 0 = W'_\infty \setminus W_\infty$. Notice that this assignment is well constructed due to condition (2). Truth constants T and F can be incorporated as special atoms satisfying $\hat{M}(T) = T_0$ and $\hat{M}(F) = F_0$, that is, $T \in W_1 \setminus W_0$ and $F \in W'_0 \setminus W'_1$.

Proposition 2. *Let* $M = (\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_\infty)$ *be a zigzag model. Then the three-valued interpretation* $\mathbf{W}_\infty$ *corresponds to collapsing all* $\hat{M}(p) = T_i$ *to true, all* $\hat{M}(p) = F_i$ *to false and* $\hat{M}(p) = 0$ *to undefined.*

It is not difficult to see that for any infinite-valued interpretation I we can always build a zigzag model M whose induced valuation coincides with I on all atoms – the next theorem asserts that it also coincides for any arbitrary formula.

Theorem 1. *Let* I *be an infinite-valued interpretation and* M *a zigzag model such that* $\hat{M}(p) = I(p)$ *for all atom* p. *Then* $I(\varphi) = \hat{M}(\varphi)$ *for any formula* φ.

Proof. We begin defining the last world $last(\varphi)$ in the chain at which formula φ does not hold so that $last(\varphi) = w_i$ when $\hat{M}(\varphi) = T_i$, $last(\varphi) = w'_{i+1}$ when $\hat{M}(\varphi) = F_i$ and $last(\varphi) = w_\infty$ when $\hat{M}(\varphi) = 0$. Note that $last(\varphi) \notin \{w'_\infty, w'_0\}$ and so $next(last(\varphi))$ (the first world at which φ holds) is always defined.

Lemma 1. *$\hat{M}(\alpha) \geq \hat{M}(\beta)$ iff $last(\alpha) \leq last(\beta)$.*

Thus, $last(\alpha) = last(\beta)$ implies $\hat{M}(\alpha) = \hat{M}(\beta)$. We proceed now by structural induction.

1. For the base case, if φ is an atom p, $\hat{M}(p) = I(p)$ by construction.
2. If $\varphi = \alpha \wedge \beta$, the last world at which φ does not hold is $max(last(\alpha), last(\beta))$. By Lemma 1 we conclude $\hat{M}(\alpha \wedge \beta) = min(\hat{M}(\alpha), \hat{M}(\beta))$. If $\varphi = \alpha \vee \beta$ the proof is analogous.
3. If $\varphi = \alpha \rightarrow \beta$, we have two cases: first, if $\hat{M}(\alpha) \leq \hat{M}(\beta)$ then, by Lemma 1, $last(\alpha) \geq last(\beta)$ and this means that any world w_k satisfies $w_k \not\models \alpha$ or $w_k \models \beta$, excepting w_0. In other words, $last(\alpha \rightarrow \beta) = w_0$ and so $\hat{M}(\alpha \rightarrow \beta) = T_0$. Otherwise, when $\hat{M}(\alpha) > \hat{M}(\beta)$, by Lemma 1 we get $last(\alpha) < last(\beta)$. Then, the last world w_k where $w_k \models \alpha$ but $w_k \not\models \beta$ is $last(\beta)$, and thus $\alpha \rightarrow \beta$ is also false until $last(\beta)$. As a result, $\hat{M}(\alpha \rightarrow \beta) = \hat{M}(\beta)$.
4. If $\varphi = {\sim}\alpha$. Assume $\hat{M}(\alpha) = T_i$, that is, $w_i \not\models \alpha$ and $w_{i+1} \models \alpha$. As $w_i = (w'_{i+1})^*$ and $w_{i+1} = (w'_{i+2})^*$ we get $w'_{i+1} \models {\sim}\alpha$ and $w'_{i+2} \not\models {\sim}\alpha$. But then, from Definition 14, we get $\hat{M}(\alpha) = F_{i+1}$. Analogously, when $\hat{M}(\alpha) = F_i$ we have $w'_{i+1} \not\models \alpha$ and $w'_i \models \alpha$ that, since $w'_{i+1} = (w_{i+2})^*$ and $w'_i = (w_{i+1})^*$, we get $w_{i+1} \not\models {\sim}\alpha$ and $w_{i+2} \models {\sim}\alpha$ and so $\hat{M}(\alpha) = T_{i+1}$. Finally, when $\hat{M}(\alpha) = 0$ we have $w_\infty \not\models \alpha$ and $w'_\infty \models \alpha$, but as $w_\infty = (w'_\infty)^*$ and $w'_\infty = (w_\infty)^*$, we obtain $w'_\infty \models {\sim}\alpha$ and $w_\infty \not\models {\sim}\alpha$ that means $\hat{M}({\sim}\alpha) = 0$. □

We can now alternatively define L_∞^{min} in terms of minimal Kripke models. Let $M_1 = (\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_\infty)$ and $M_2 = (\mathbf{U}_0, \mathbf{U}_1, \ldots, \mathbf{U}_\infty)$ be a pair of zigzag models. We say that $M_1 \trianglelefteq M_2$ iff either $M_1 = M_2$ or $\exists i \left(\forall j \ (j \leq i \Rightarrow \mathbf{W}_j = \mathbf{U}_j) \wedge \mathbf{W}_i \prec \mathbf{U}_i\right)$.

Proposition 3. *$M_1 \trianglelefteq M_2$ iff $\hat{M}_1 \sqsubseteq_\infty \hat{M}_2$.*

Therefore, we have two alternative but equivalent definitions of the semantics of disjunctive logic programs with negation. In the rest of the paper, the two approaches will be used interchangeably.

4 Existence of Minimal Models

In this section we demonstrate that every disjunctive program has at least one minimal infinite-valued model. The proof is based on the dual of Zorn's Lemma[5]:

Lemma 2 (The dual of Zorn's Lemma). *Every non-empty partially ordered set in which each downward chain has a lower bound, contains a minimal element.*

[5] Notice that the proof given in this section can be extended to apply to infinite propositional programs (and therefore also to the case of first-order programs).

First, notice that the set of models of a disjunctive logic program is non-empty, because the interpretation which assigns to every propositional atom the value T_0 is always a model. Second, notice that the set of models of a program is partially ordered by the $\sqsubseteq_\infty$ relation. It suffices to show that every (possibly transfinite) downwards chain of models under $\sqsubseteq_\infty$, has a lower bound which is also a model of the program.

Therefore, consider a chain $\mathcal{M}$ of infinite-valued models of a disjunctive program Π: $M_0 \sqsupseteq_\infty M_1 \sqsupseteq_\infty M_2 \sqsupseteq_\infty \cdots \sqsupseteq_\infty M_\alpha \sqsupseteq_\infty \cdots$. We describe at an intuitive level the construction of a lower bound M of this chain. We first start with all models that belong to $\mathcal{M}$ and we "intersect" them at their zero'th level of truth values. This gives us a (partial) interpretation that will serve as the zero'th level of the lower bound M. We now consider only those elements of the chain $\mathcal{M}$ whose zero'th order part agrees with the partial interpretation we have just constructed. We repeat the above process with this new set of models and with the order one values. In the limit of this process, certain atoms may have not received a value; we assign to them the value 0. We now formalize the above construction:

Definition 15. *Let S be a set of infinite-valued interpretations of a given program and $n \in \omega$. Then, we define $\bigwedge^n S = \{(p, T_n) \mid \forall M \in S, M(p) = T_n\}$ and $\bigvee^n S = \{(p, F_n) \mid \exists M \in S, M(p) = F_n\}$. Moreover, we define $\bigodot^n S = (\bigwedge^n S) \bigcup (\bigvee^n S)$.*

Let Π be a program and let $\mathcal{M}$ be a downward chain of models of Π. We can now define the following sequence of sets of models of Π:

$$\begin{aligned} S_0 &= \mathcal{M} \\ S_{n+1} &= \{M \in S_n \mid M \sharp n = \bigodot^n S_n\} \end{aligned}$$

We now have the following lemma:

Lemma 3. *For every $n < \omega$, $S_n \neq \emptyset$.*

Proof. We demonstrate by induction on n that S_n is a non-empty chain identical to $\mathcal{M}$ the only difference being that an initial part of $\mathcal{M}$ may be missing from S_n. The base case is obvious. Assume the statement holds for n ie., that S_n is a nonempty downward chain of the form $M_\alpha \sqsupseteq_\infty M_{\alpha+1} \sqsupseteq_\infty M_{\alpha+2} \sqsupseteq_\infty \cdots$. For the induction step observe that since $M_\alpha \parallel T_n \supseteq M_{\alpha+1} \parallel T_n \supseteq \cdots$ and $M_\alpha \parallel F_n \subseteq M_{\alpha+1} \parallel F_n \subseteq \cdots$, after an initial segment of this chain, all the members of the chain agree on their level n components. Therefore, S_{n+1} forms a non-empty chain. This establishes the desired result. □

Example 2. Consider the program just consisting of rule: $s \vee p \leftarrow\sim s$

Moreover, consider the following models of the above program:
$M_n = \{(s, T_n), (p, F_0)\}$. Clearly, $M_0 \sqsupseteq_\infty M_1 \sqsupseteq_\infty M_2 \sqsupseteq_\infty \cdots$. Then, it is not hard to see that $S_0 = \{M_0, M_1, M_2, M_3, \ldots\}$, $S_1 = \{M_1, M_2, M_3, \ldots\}$, and so on.

We can now demonstrate the main theorem of this section which actually states that for every downward chain of models of a given disjunctive program, there exists a lower bound:

Theorem 2. *Let Π be a program and let $\mathcal{M}$ be a downwards chain of models of Π. Then $\mathcal{M}$ has a lower bound which is a model of Π.*

Proof. Consider models $N_0 \in S_1, N_1 \in S_2, \ldots, N_k \in S_{k+1}, \ldots, k < \omega$. We construct an interpretation M as follows:

$$M(p) = \begin{cases} (\bigcup_{k<\omega}(N_k \sharp k))(p) & \text{if this is defined} \\ 0 & \text{otherwise} \end{cases}$$

We claim that M is a model of the program. Assume it is not. Consider then a clause $p_1 \vee \cdots \vee p_m \leftarrow B$ such that $M(p_1 \vee \cdots \vee p_m) < M(B)$. There are three cases:

- $M(p_1 \vee \cdots \vee p_m) = F_k$, $k < \omega$. But then, $N_k(p_1 \vee \cdots \vee p_m) = F_k$ and since N_k is a model of Π, we have $N_k(B) \leq F_k$. But this implies that $M(B) \leq F_k$, and therefore $M(p_1 \vee \cdots \vee p_m) \geq M(B)$ (contradiction).
- $M(p_1 \vee \cdots \vee p_m) = T_k$, $k < \omega$. Then, $N_k(p_1 \vee \cdots \vee p_m) = T_k$ and since N_k is a model of Π, we have $N_k(B) \leq T_k$. But this easily implies that $M(B) \leq T_k$, and therefore $M(p_1 \vee \cdots \vee p_m) \geq M(B)$ (contradiction).
- $M(p_1 \vee \cdots \vee p_m) = 0$. But then, for every $k < \omega$, we have $N_k(p_1 \vee \cdots \vee p_m) < T_k$. Now, since N_k is a model of Π for every k, it is $N_k(B) < T_k$. But this then implies that $M(B) \leq 0$, and therefore $M(p_1 \vee \cdots \vee p_m) \geq M(B)$ (contradiction).

Moreover, it is straightforward to see that, by construction, M is a lower bound for all the members of the chain. □

The above discussion leads to the following theorem:

Theorem 3. *Every disjunctive logic program with negation has a non-empty set of minimal infinite-valued models.*

Proof. Immediate from the dual of Zorn's Lemma and Theorem 2. □

Example 3. Consider again the program:

$$s \vee p \leftarrow\sim s$$

together with the models:

$$M_n = \{(s, T_n), (p, F_0)\}$$

Applying the above construction to the chain $M_0 \sqsupseteq_\infty M_1 \sqsupseteq_\infty M_2 \sqsupseteq_\infty \cdots$, we get the lower bound $M = \{(s, 0), (p, F_0)\}$ of the chain.

Actually, it is not hard to see that the above program has two minimal models, namely $\{(p, T_1), (s, F_0)\}$ and $\{(p, F_0), (s, 0)\}$.

Example 4. Consider the program:

$$\begin{array}{ll} p \vee q \vee r \leftarrow & \\ p & \leftarrow \sim q \\ q & \leftarrow \sim r \\ r & \leftarrow \sim p \end{array}$$

By inspection, this program has the three minimal models $\{(p, T_0), (q, T_2), (r, F_1)\}$, $\{(p, F_1), (q, T_0), (r, T_2)\}$ and $\{(p, T_2), (q, F_1), (r, T_0)\}$.

5 Properties of the Minimal Model Semantics

We turn to properties of the minimal infinite-valued semantics. First we see that new approach extends the well-founded semantics of normal logic programs:

Proposition 4. *A normal logic program Π has a $\trianglelefteq$-minimum model $M = (\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_\infty)$ where $\mathbf{W}_\infty$ is the well-founded model of Π.*

Proof. [9] shows that every normal logic program has a minimum infinite-valued model (Theorem 7.4) which when collapsed to three-valued logic coincides with the well-founded model (Theorem 7.6). From these two results and Proposition 2, the result above follows immediately. □

In other words, when we restrict the syntax to that of normal logic programs, the infinite-valued approach provides the well-founded model of the program, apart from additional information. In the case of the $\sqsubseteq_\infty$-least model characterisation, the well-founded model is obtained by collapsing all the T_i and F_i values in the model into T and F respectively. In the case of the corresponding $\trianglelefteq$-least zigzag model, the well-founded model is *directly obtained* by just keeping the $\mathbf{W}_\infty$ pair and ignoring all the rest. In fact, when we later compare the infinite-valued approach to other disjunctive well-founded semantics, we will also restrict the study to pair $\mathbf{W}_\infty$ so that we just handle a three-valued interpretation. A different and interesting open topic is the possible utility of the rest of information contained in the infinite-valued minimal models, which captures somehow the ordering or level in which we make default assumptions to compute the final result.

As we are now going to see, the approach we propose is compatible with the minimal model semantics for (negation-less) disjunctive logic programs. A *positive disjunctive* logic program is a set of clauses like (1) where $m = 0$ (i.e., they have no negated literals). Given a classical model X we can define a corresponding 1-complete zigzag model M^X so that $\mathbf{W}_1 = (X, X)$. This implies all $W_j = W'_j = X$ except $W_0 = \emptyset$ and $W'_0 = Q$ that are fixed. An i-complete model assigns to any atom a value of order smaller than i – when $i = 1$ the value can just be T_0 or F_0. The following pair of lemmas can be easily proved.

Lemma 4. *For a positive disjunctive program Π, $X \models \Pi$ in classical logic iff $M^X \models \Pi$.*

Lemma 5. *Any $\trianglelefteq$-minimal model of a positive disjunctive program Π is 1-complete.*

Theorem 4. *Let Π be a positive disjunctive logic program. Then: (i) if X is a minimal classical model of Π then M^X is a $\trianglelefteq$-minimal model of Π; (ii) if $M = (\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_\infty)$ is a $\trianglelefteq$-minimal model of Π, then M is 1-complete, and W_1 is a minimal classical model of Π.*

Proof. (i) If X is a minimal classical model, by Lemma 4, $M^X \models \Pi$. Assume we have some $\trianglelefteq$-minimal model of Π strictly smaller than M^X – by Lemma 5, such a model is 1-complete, call it M^Y, and since it is strictly smaller than M^X, $Y \subset X$. By Lemma 4, $Y \models \Pi$, and this contradicts minimality of classical model X. For (ii), Lemma 5 directly implies that M is 1-complete, i.e., $\mathbf{W}_1 = (W_1, W_1)$. By Lemma 4, $W_1 \models \Pi$. Assume there exists a smaller classical model $Y \subset W_1$. By Lemma 4, $M^Y \models \Pi$. Since M^Y is $\trianglelefteq$-smaller than M^X we get a contradiction. □

6 Identifying the Minimal Models

In this section we demonstrate that every disjunctive logic program has a *finite* set of minimal infinite-valued models. This result is not immediately obvious since the underlying logic has an infinite number of truth values. The key idea behind the proof is that if $|S|$ is the number of propositional symbols of a program, then it suffices to consider as possible candidates for minimality the models of the program that use truth values with order at most $|S| - 1$. The following definition will be needed in the proof of the theorem:

Definition 16. *Let Π be a program and let M be an infinite-valued model of Π. We will say that M* contains a gap at order $\delta \in \omega$ *if:*

- *For every $n < \delta$, there exists an atom p in Π with $order(M(p)) = n$.*
- *There does not exist an atom p in Π with $order(M(p)) = \delta$.*
- *There exists an atom p of Π such that $\delta < order(M(p)) < \infty$.*

The proof of the following theorem demonstrates that minimal models cannot contain gaps. This easily implies that in our search for minimal models we need only inspect a finite number of models.

Theorem 5. *Let Π be a program, S be the set of propositional symbols that appear in P and let M be a minimal infinite-valued model of Π. Then, for every propositional symbol $p \in S$, $M(p) \in \{0, F_0, T_0, \ldots, F_{|S|-1}, T_{|S|-1}\}$.*

Proof. It suffices to demonstrate that if a model of Π contains a gap then it can not be minimal. The theorem then follows by the fact that if a program does not contain a gap then its propositional symbols will necessarily get values from the set $\{0, F_0, T_0, \ldots, F_{|S|-1}, T_{|S|-1}\}$.

Assume for the sake of contradiction that M is a minimal model of Π that contains a gap at order $\delta \in \omega$. We establish a contradiction by constructing a model M^* of Π such that $M^* \sqsubset M$. Let $m > \delta$ be the least natural number such $M\sharp m \neq \emptyset$. We distinguish the following two cases:

Case 1: There exists some p such that $(M\sharp m)(p) = T_m$. We define the following interpretation:

$$M^*(p) = \begin{cases} T_{m+1}, \text{ if } M(p) = T_m \\ M(p), \text{ otherwise} \end{cases}$$

Obviously, $M^* \sqsubset M$. Let $p_1 \vee \cdots \vee p_n \leftarrow B$ be a clause in Π. By a simple case analysis on the possible values of $M^*(p_1 \vee \cdots \vee p_n)$, we get that M^* satisfies the given clause and therefore the whole program. Consequently, M is not a minimal model of Π (contradiction).

Case 2: There does not exist any p such that $(M\sharp m)(p) = T_m$. We define the following interpretation:

$$M^*(p) = \begin{cases} T_{n-1}, \text{ if } M(p) = T_n \text{ and } n > \delta \\ F_{n-1}, \text{ if } M(p) = F_n \text{ and } n > \delta \\ M(p), \text{ otherwise} \end{cases}$$

Obviously, $M^* \sqsubset M$. Let $p_1 \vee \cdots \vee p_n \leftarrow B$ be a clause in Π. By a simple case analysis on the possible values of $M^*(p_1 \vee \cdots \vee p_n)$, we get that M^* satisfies the given clause and therefore the whole program. Consequently, M is not a minimal model of Π (contradiction). Therefore, if a model of Π contains a gap, it cannot be a minimal one. □

7 Related Approaches

In this section we mention some examples that are useful for comparing the infinite-valued approach L_∞^{min} with other existing approaches to the semantics of disjunctive logic programs with negation, in particular, STATIC (of [8]), D-WFS (of [2, 3]), WFDS (of [11]), WFS_d of [1] and with PEL (of [4]). We observe in particular that L_∞^{min} differs from all these semantics.

Example 1 was presented in [11], where it was reasoned that b should be false, while STATIC and D-WFS just fail to derive any information. In fact, the three semantics WFDS, PEL and L_∞^{min} allow one to derive the falsity of b.

Consider now the following example borrowed from [1]: $\{(a \vee b \leftarrow\sim b), (b \leftarrow\sim b)\}$. For this example, L_∞^{min} yields the minimal model $\{(a, F_0), (b, 0)\}$. PEL agrees that a should be false and b undefined. However, WFDS makes both atoms undefined.

An interesting observation is that, as in PEL, the *unfolding* transformation rule (see eg [3]) does not preserve equivalence in L_∞^{min}. Unfolding atom b on program $\Pi_1 = \{(a \vee b), (a \leftarrow\sim a), (c \leftarrow a \wedge b)\}$ leads to program $\Pi_2 = \{(a \vee b), (a \leftarrow\sim a), (c \vee a \leftarrow a)\}$. Both programs have the minimal model $\{(a, T_0), (b, F_0), (c, F_0)\}$ but Π_1 has a second minimal model $\{(a, 0), (b, T_0), (c, 0)\}$ while the second minimal model of Π_2 is $\{(a, 0), (b, T_0), (c, F_0)\}$ (in fact, PEL agrees with these results too). It follows that L_∞^{min} differs from WFS_d ([1]).

Another similarity between PEL and the L_∞^{min} semantics is that applying the *S-Implication* (S-IMP) transformation rule from WFDS [12] does not generally yield a strongly equivalent program. For instance, the result of applying S-IMP on program $\Pi_3 = \{(b \vee c \leftarrow a), (b \leftarrow a \wedge \sim c)\}$ deletes the second rule $\Pi_4 = \{(b \vee c \leftarrow a)\}$. However, if we add $\Pi_5 = \{(c \leftarrow\sim a), (a \leftarrow\sim a)\}$ to both programs, we obtain that $\Pi_3 \cup \Pi_5$ and $\Pi_4 \cup \Pi_5$ have different L_∞^{min} models: both have unique minimal models, but the former makes all atoms undefined, while the latter makes b false and the rest undefined.

Partial equilibrium logic (PEL) ([4]) is a general nonmonotonic framework that extends the partial stable model semantics of [7]. In some respects it appears to be conceptually close to the semantics of L_∞^{min}. In particular, it also provides a purely declarative, model-theoretic semantics and is even based on Routley frames. However, the most important difference has to do with the existence of L_∞^{min} model for disjunctive programs, something not guaranteed by PEL. This is illustrated by Example 4, which has no PEL models whereas, as we saw before, it has three L_∞^{min} models. Another difference is that in the L_∞^{min} approach the intended models are reached via one minimization process. In PEL one first defines partial stable or partial equilibrium models through a minimization process, while a second minimality condition captures those models that are said to be well-founded.

8 Conclusions

We have introduced a new purely model-theoretic semantics for disjunctive logic programs with negation and showed that every such program has at least one minimal infinite-valued model. The new semantics generalizes both the minimal model semantics of positive disjunctive logic programs as well as the well-founded semantics of normal logic programs. Future work includes the study of efficient proof procedures for the new semantics and possible applications of the new

approach. An interesting open question is the possible application of the additional information provided by the infinite-valued approach not present in other variants of well-founded semantics which return three-valued answers. This extra information is related to the level or ordering in which default assumptions are made to compute the final result, and can be of valuable help for debugging an unexpected outcome, pointing out unobserved dependences or even capturing priorities as different truth levels. We also plan to investigate the underlying logic L_∞ of this approach in more detail. Another interesting topic for future research is the generalization of the recently introduced *game semantics* of negation [5] to the case of disjunctive logic programs.

References

1. J. Alcantara, C.V. Damasio and L.M. Pereira. A Well-Founded Semantics with Disjunction. In *Proceedings of the International Conference on Logic Programming (ICLP'05)*, LNCS 3668, pages 341–355, 2005.
2. S. Brass and J. Dix. Characterizations of the (disjunctive) Stable Semantics by Partial Evaluation. *Journal of Logic Programming*, 32(3):207–228, 1997.
3. S. Brass and J. Dix. Characterizations of the Disjunctive Well-founded Semantics: Confluent Calculi and Iterated GCWA. *Journal of Automated Reasoning*, 20(1):143–165, 1998.
4. P. Cabalar, S. Odintsov, D. Pearce, and A. Valverde. Analysing and Extending Well-Founded and Partial Stable Semantics using Partial Equilibrium Logic. In *Proceedings of the International Conference on Logic Programming (ICLP'06)*, LNCS 4079, pages 346–360, Seattle, USA, August 2006.
5. Ch. Galanaki, P. Rondogiannis and W.W. Wadge. An Infinite-Game Semantics for Well-Founded Negation. *Annals of Pure and Applied Logic*, 2007 (to appear).
6. D. Pearce. Equilibrium Logic. *Ann. Math & Artificial Int.*, 47 (2006), 3-41.
7. Przymusinski, T. Well-founded and stationary models of logic programs. *Annals of Mathematics and Artificial Intelligence* 12:141–187, 1994.
8. T. Przymusinski. Static Semantics of Logic Programs. *Annals of Mathematics and Artificial Intelligence*, 14(2-4):323–357, 1995.
9. P. Rondogiannis and W.W. Wadge. Minimum Model Semantics for Logic Programs with Negation-as-Failure. *ACM Transactions on Computational Logic*, 6(2):441–467, 2005.
10. A. van Gelder, K. A. Ross, and J. S. Schlipf. The Well-Founded Semantics for General Logic Programs. *Journal of the ACM*, 38(3):620–650, 1991.
11. K. Wang. Argumentation-based Abduction in Disjunctive Logic Programming. Journal of Logic Programming, 45(1-3):105–141, 2000.
12. K. Wang and L. Zhou. Comparisons and Computation of Well-Founded Semantics for Disjunctive Logic Programs. ACM Transactions on Computational Logic, 6(2):295–327, 2005.